

1 What is claimed is:

2

3 1. In a data processing network including distributed processing units, a
4 method comprising:

5 obtaining a respective utilization value of each distributed processing unit;

6 applying a mapping function to the respective utilization value of said each
7 distributed processing unit to obtain a respective weight for said each distributed
8 processing unit; and

9 using the respective weights for the distributed processing units for distributing
10 work requests to the distributed processing units so that the respective weight for said
11 each distributed processing unit specifies a respective frequency at which the work
12 requests are distributed to said each distributed processing unit.

13

14 2. The method as claimed in claim 1, wherein the respective utilization value
15 of said each distributed processing unit is a percentage of saturation of said each
16 distributed processing unit.

17

18 3. The method as claimed in claim 1, wherein said each distributed
19 processing unit collects statistics for calculation of the respective utilization value of said
20 each distributed processing unit.

21

1 4. The method as claimed in claim 1, wherein statistics for calculation of the
2 respective utilization value of said each distributed processing unit are collected from
3 said each distributed processing unit.
4

5 5. The method as claimed in claim 1, wherein the respective weight for said
6 each distributed processing unit is programmed into a mapping table, and the mapping
7 function is applied to the respective utilization value of said each distributed processing
8 unit to obtain the respective weight for said each distributed processing unit by indexing
9 the mapping table with the respective utilization value of said each distributed processing
10 unit to obtain the respective weight for said each distributed processing unit.
11

12 6. The method as claimed in claim 1, wherein the mapping function is
13 selected to provide weights estimated to cause a balancing of loading upon the distributed
14 processing units.
15

16 7. The method as claimed in claim 1, wherein the respective weights are used
17 for weighted round-robin load balancing of the work requests upon the distributed
18 processing units.
19

20 8. The method as claimed in claim 7, wherein the weighted round-robin load
21 balancing performs round-robin load balancing when the weights are equal.
22

1 9. The method as claimed in claim 1, wherein the respective weights for the
2 distributed processing units are used for distributing work requests to the distributed
3 processing units by creating a distribution list containing entries indicating the distributed
4 processing units, the respective weight for said each distributed processing unit
5 specifying the number of the entries indicating said each distributed processing unit, and
6 by randomizing the distribution list, and accessing the randomized distribution list for
7 distributing the work requests to the distributed processing units as indicated by the
8 entries in the randomized distribution list.

9

10 10. The method as claimed in claim 1, which includes re-randomizing the
11 distribution list for re-use once the end of the distribution list is reached during the
12 distribution of the work requests to the distributed processing units as indicated by the
13 entries in the randomized distribution list.

14

15 11. In a data processing network including distributed processing units, a
16 method comprising:

17 obtaining a respective utilization value of each distributed processing unit;

18 applying a mapping function to the respective utilization value of said each
19 distributed processing unit to obtain a respective weight for said each distributed
20 processing unit;

21 using the respective weights for the distributed processing units for producing a
22 distribution list for distributing work requests to the distributed processing units for load
23 balancing of the work requests upon the processing units, and

1 repetitively randomizing the distribution list during the distribution of the work
2 requests to the distributed processing units.

3
4

5 12. In a data processing network including a network file server and a
6 plurality of virus checking servers, a method comprising:

7 the network file server obtaining a respective utilization value of each virus
8 checking server, the respective utilization value of said each virus checking server
9 indicating a percentage of saturation of said each virus checking server;

10 the network file server applying a mapping function to the respective utilization
11 value of said each virus checking server to obtain a respective weight for said each virus
12 checking server; and

13 the network file server using the respective weights for the virus checking servers
14 for weighted round-robin load balancing of virus checking requests from the network file
15 server to the virus checking servers.

16

17 13. The method as claimed in claim 12, wherein said each virus checking
18 server collects statistics for calculation of the respective utilization value of said each
19 virus checking server.

20

21 14. The method as claimed in claim 12, wherein the respective weight for said
22 each virus checking server is programmed into a mapping table, and the network file
23 server indexes the mapping table with said each respective utilization value to obtain the
24 respective weight for said each virus checking server.

1

2 15. The method as claimed in claim 12, wherein the weighted round-robin
3 load balancing performs round-robin load balancing when the weights are equal.

4

5 16. The method as claimed in claim 12, wherein the respective weights for the
6 virus checking servers are used for weighted round-robin load balancing of virus
7 checking requests from the network file server to the virus checking servers by creating a
8 distribution list containing entries indicating the virus checking servers, the respective
9 weight for said each virus checking server specifying the number of the entries indicating
10 said each virus checking server, and by randomizing the distribution list, and accessing
11 the randomized distribution list for distributing the virus checking requests from the
12 network file server to the virus checking servers as indicated by the entries in the
13 randomized distribution list.

14

15 17. The method as claimed in claim 16, which includes re-randomizing the
16 distribution list for re-use once the end of the distribution list is reached during the
17 distributing of the work requests to the virus checking servers as indicated by the entries
18 in the randomized distribution list.

19

20 18. The method as claimed in claim 16, wherein the network file server
21 obtains the utilization values of the virus checking servers at the start of a heartbeat
22 interval, randomizes the distribution list repetitively during use of the distribution list for
23 load balancing of virus checking requests during the heartbeat interval, obtains new

1 utilization values of the virus checking servers at the start of a following heartbeat
2 interval, and produces a new distribution list from the new utilization values of the virus
3 checking servers for load balancing of virus checking requests during the following
4 heartbeat interval.

5

6 19. A data processing system comprising distributed processing units and a
7 processor coupled to the distributed processing units for distributing work requests to the
8 distributed processing units, the processor being programmed for:

9 obtaining a respective utilization value of each distributed processing unit;

10 applying a mapping function to the respective utilization value of said each
11 distributed processing unit to obtain a respective weight for said each distributed
12 processing unit; and

13 using the respective weights for the distributed processing units for distributing
14 work requests to the distributed processing units so that the respective weight for said
15 each distributed processing unit specifies a respective frequency at which the work
16 requests are distributed to said each distributed processing unit.

17

18 20. The data processing system as claimed in claim 19, wherein the respective
19 utilization value of said each distributed processing unit is a percentage of saturation of
20 said each distributed processing unit.

21

1 21. The data processing system as claimed in claim 19, wherein said each
2 distributed processing unit is programmed for collecting utilization statistics of said each
3 distributed processing unit.

4
5 22. The data processing system as claimed in claim 19, wherein the processor
6 is programmed for collecting utilization statistics from said each distributed processing
7 unit.

8
9 23. The data processing system as claimed in claim 19, wherein the respective
10 weight for said each distributed processing unit is programmed into a mapping table, and
11 the processor is programmed to apply the mapping function to the respective utilization
12 value of said each distributed processing unit to obtain a respective weight for said each
13 distributed processing unit by indexing the mapping table with said each respective
14 utilization value of said each distributed processing unit to obtain the respective weight
15 for said each distributed processing unit.

16
17 24. The data processing system as claimed in claim 19, wherein the mapping
18 function is programmed to produce weights estimated to cause a balancing of loading
19 upon the distributed processing units.

20
21 25. The data processing system as claimed in claim 19, wherein the processor
22 is programmed for using the respective weights for weighted round-robin load balancing
23 of the work requests upon the distributed processing units.

1

2 26. The data processing system as claimed in claim 19, wherein the processor
3 is programmed for performing round-robin load balancing of the work requests upon the
4 distributed processing units when the weights are equal.

5

6 27. The data processing system as claimed in claim 19, wherein the processor
7 is programmed for using the respective weights for the distributed processing units for
8 distributing work requests to the distributed processing units by creating a distribution list
9 containing entries indicating the distributed processing units, the respective weight for
10 said each distributed processing unit specifying the number of the entries indicating said
11 each distributed processing unit, and by randomizing the distribution list, and accessing
12 the randomized distribution list for distributing the work requests to the distributed
13 processing units as indicated by the entries in the randomized distribution list.

14

15 28. The data processing system as claimed in claim 19, wherein the processor
16 is programmed for re-randomizing the distribution list for re-use once the end of the
17 distribution list is reached during the distribution of the work requests to the distributed
18 processing units as indicated by the entries in the randomized distribution list.

19

20 29. A data processing system comprising distributed processing units and a
21 processor coupled to the distributed processing units for distributing work requests to the
22 distributed processing units, the processor being programmed for:

23 obtaining a respective utilization value of each distributed processing unit;

1 applying a mapping function to the respective utilization value of said each
2 distributed processing unit to obtain a respective weight for said each distributed
3 processing unit;

4 using the respective weights for the distributed processing units for producing a
5 distribution list for distributing work requests to the distributed processing units for load
6 balancing of the work requests upon the processing units, and

7 repetitively randomizing the distribution list during the distribution of the work
8 requests to the distributed processing units.

9
10 30. A data processing system comprising virus checking servers and a
11 network file server coupled to the virus checking servers for distributing virus checking
12 requests to the virus checking servers, the network file server being programmed for:

13 obtaining a respective utilization value of each virus checking server, the
14 respective utilization value of said each virus checking server indicating a percentage of
15 saturation of said each virus checking server;

16 applying a mapping function to the respective utilization value of said each virus
17 checking server to obtain a respective weight for said each virus checking server; and

18 using the respective weights for the virus checking servers for weighted round-
19 robin load balancing of virus checking requests from the network file server to the virus
20 checking servers.

21

1 31. The data processing system as claimed in claim 30, wherein said each
2 virus checking server is programmed for collecting statistics for calculating the respective
3 utilization value of said each virus checking server.

4
5 32. The data processing system as claimed in claim 30, wherein the respective
6 weight for said each virus checking server is programmed into a mapping table, and the
7 network file server is programmed for indexing the mapping table with said each
8 respective utilization value of said each virus checking server to obtain the respective
9 weight for said each virus checking server.

10
11 33. The data processing system as claimed in claim 30, wherein the network
12 file server is programmed for performing round-robin load balancing of the virus
13 checking requests upon the virus checking servers when the weights are equal.

14
15 34. The data processing system as claimed in claim 30, wherein the network
16 file server is programmed for using the respective weights for the virus checking servers
17 for weighted round-robin load balancing of virus checking requests from the network file
18 server to the virus checking servers by creating a distribution list containing entries
19 indicating the virus checking servers, the respective weight for said each virus checking
20 server specifying the number of the entries indicating said each virus checking server,
21 and by randomizing the distribution list, and accessing the randomized distribution list
22 for distributing the virus checking requests from the network file server to the virus
23 checking servers as indicated by the entries in the randomized distribution list.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

35. The data processing system as claimed in claim 34, wherein the network file server is programmed for re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus checking servers as indicated by the entries in the randomized distribution list.

36. The data processing system as claimed in claim 34, wherein the network file server is programmed for collecting utilization statistics from the virus checking servers at the start of a heartbeat interval, for randomizing the distribution list repetitively during use of the distribution list for load balancing of virus checking requests during the heartbeat interval, for collecting a new set of utilization statistics from the virus checking servers at the start of a following heartbeat interval, and for producing a new distribution list from the new set of utilization statistics for load balancing of virus checking requests during the following heartbeat interval.